ATLANTIS PRESS

Advances in Social Science, Education and Humanities Research, volume 196

2nd International Conference on Social Science, Public Health and Education (SSPHE 2018)

Reviewer Assignment Strategy of Peer Assessment: Towards Managing Collusion in Self-assignment

Yanqing Wang*
School of Economics and Management
Harbin Institute of Technology
Heilongjiang, China
yanqing@hit.edu.cn

Bingyu Liu
School of Mathematics and Statistics
Northeastern University at Qinhuangdao
Hebei, China
liuby@neuq.edu.cn

Kun Zhang
School of Mathematics and Statistics
Northeastern University at Qinhuangdao
Hebei, China
zhangkun@neuq.edu.cn

Yushan Jiang
School of Mathematics and Statistics
Northeastern University at Qinhuangdao
Hebei, China
jys@neuq.edu.cn

Fuquan Sun
School of Mathematics and Statistics
Northeastern University at Qinhuangdao
Hebei, China
sunfq@neuq.edu.cn

Abstract—Peer assessment is an efficient and effective learning process that has been widely used in diverse fields in the higher education. Despite of its many benefits, collusion is a common challenge that makes the reliability of peer assessment a primary concern in practices, especially when self-assignment strategy is applied in reviewer assignment. This research aims to propose a model of collusion management that applies co-occurrence network as a means to identify collusion. The proposed model is implemented as a software module in a peer code review system called EduPCR. EduPCR is able to monitor this measure and trigger instructor's inquiries to collusion suspects when it identifies suspected collusion. A verification implemented in a university-level C Programming course shows that the proposed collusion management model is reasonable and the identification algorithm is practical in diverse peer assessment contexts.

Keywords—peer assessment, reviewer assignment, collusion, co-occurrence network, self-assignment strategy, EduPCR

I. INTRODUCTION

Study of peer assessment (PA), also named peer review or peer evaluation alternatively, has a long history and can be traced back to 1970s [1, 2]. Due to its efficiency and active learning nature, PA has been widely used in diverse fields [3, 4] for collaborative learning. In the study on undergraduate peer assessments, students reported positive experience in peer assessment and recommended its use in college courses [5]. Also, the necessity of peer assessment in writing learning was claimed [6]. However, besides many benefits from peer assessment, there are still common pitfalls for collaborative learning involving peer assessment, such as mutual admiration societies and inaccuracy assessment [7]. These pitfalls were resulted in a common phenomenon, i.e. collusion.

Li claims that, in a mutual admiration society, everyone is a peer and everyone respects each other. This doesn't mean that everyone will agree on everything. It just means that disagreements are respectful [8]. The reasons that mutual admiration society is considered as a pitfall is that the students might miss-apply the principles of a mutual admiration society and try to avoid taking off other students' marks with an expectation that the other students won't take off their own marks either [8].

Due to the close relationship between students' final academic scores and evaluation results given by their peers, collusion which appears in peer assessment would be a problem. The current General Information Catalog of the University of Texas at Austin defines collusion as "unauthorized collaboration with another person in preparing academic assignments offered for credit or collaboration with another person to commit a violation of any section of the rules on scholastic dishonesty." In online PA systems, if students know game rules of a reviewer assignment strategy, they may collude to give high-scores (we regard high-score as an unreasonable and high one in this study) within a small range to each other to receive high code score, high review score [9]. The categorization of collusion in PA is divided into small-circle collusion and pervasive collusion [10].

Actually, collusion is likely to happen when anonymity strategy is not applied in reviewer assignment of PA, i.e. whenever students can recognize their peers' identity, collusion tends to exist. Thus, the anonymity of reviewer assignment is closely related to collusion. Moreover, who is the assigner of reviewers affects the anonymity of reviewer assignment in PA.

A. Anonymity of Reviewer Assignment in PA

To avoid unfair ratings by known peers, anonymity strategy is widely applied in PA. Anonymity is an important part of the scientific method, used to prevent research outcomes from being "influenced" by the placebo effect or observer bias. Blank finds that the reviewers are more critical when they are unaware of the author's identity [11]. In double-blinded strategy, a student does not know peer's identity, which ensures the quality of the review by excluding the factors of human relationships. Moreover, the problem of ghostwriting could be inhibited to a great extent. The participants in PA should be anonymous. Ballantyne et al. report that anonymous peer assessment reduces the opportunity of collusion and biased marking [12].

However, since PA happens in a learning society rather than a set of machines, double-blinded strategy is not always optimal. For example, single-blinded (either defined as authors'

National Natural Science Foundation of China (71573065, 71571085), MOE of China (201602016006), and Hebei Dept. of Education (2017GJXGK028)

Copyright © 2019, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

blindness to reviewer's identity or reviewers' blindness to authors' identity) is a practical reviewer assignment strategy. Especially, a non-anonymity (real-name completely) strategy, a union of single-blinded to author and single-blinded to reviewer, might make a learning environment active, challenging, and different.

B. Assigner of Reviewers in PA

Besides anonymity policy in PA, there is another relevant decision, i.e. who assigns reviewer to author's work. It is critical because it affects both quality control and participants' passion for learning. The assigner of reviewer assignment in PA can be student, instructor, or computer [13]. Among these three strategies, permitting student as assigner (self-assignment) is an interesting choice. It is a special case of non-anonymity, which enables students to choose favorite reviewers so that author knows the identities of reviewers at least. It is also named as free-selection protocol (students are allowed to freely explore and select peer work for review) [14]. It is indicated that students who follow the free-selection protocol will demonstrate better domain learning outcomes and better reviewer skills than those who follow the assigned-protocol (the instructor assigns student works for review to student pairs) condition [14]. Also, a self-assignment strategy is implemented, in which students choose their reviewers by themselves [8].

The self-assignment strategy might be debatable. In that circumstance, capable students will be popular and will be chosen by more than one students. On the one hand, it may cause problems such as unfair distribution (assignment) and collusion. On the other hand, it may prompt challenge and play a role of motivation (early birds get food) [13].

In summary of non-anonymity and self-assignment, the potential advantages of self-assignment strategy includes [13]:

1) For author: Knowing reviewers' name helps author shield against disturbance caused by the misunderstanding of reviewers' suggestions or mistakes made by reviewers. Also, if self-assignment strategy is applied, a motivation effect could be achieved, i.e. early authors (they submit their work earlier than others) get excellent (or favorite) reviewers.

2) For reviewer: Getting authors' names allows reviewer shield against bewilderment of authors' advanced artifacts (work to be reviewed). Moreover, non-anonymity may boost efficient and effective learning. For one example, a reviewer who admires his/her author might pay more attention to reflect and imitate. Also, if a reviewer is familiar to an author, it may save his/her time to understand the author's work.

Thus, if a scholar or instructor has the courage to attempt self-assignment, the next critical problem is how to manage the collusion or how to design a practical management policy.

To solve the collusion management problem, the article is organized as follows. Background section presents two examples of online PA system, collusion categorization, and definitions in this study. Methodology section embraces a model of collusion management and its key algorithm. Implementation section explores the feasibility of the collusion identification algorithm. Finally, discussions are presented and future research directions are pointed out.

II. Background

A. Examples of Online PA Systems

1) EduPCR (peer code review in educational context): It is an online PA system for programming language learning. It is suitable for undergraduates, postgraduates, and junior college students who need to learn programming skills. In EduPCR, every learner is required to participate in some phases, such as completing a coding task, reviewing peers' work, revising own program, doing back-evaluation, etc. Instructor is responsible for setting tasks and summarizing final scores of students. Computer assigns reviewers and manages the schedule [15].

Since 2004, EduPCR has been updated for several versions. It was applied in the evaluation process of three courses, including C++, C Programming, and Object-Oriented Programming in Java, at two schools of Harbin Institute of Technology. With this open-type learning approach, students improved greatly in their high-order capabilities, such as analyzing, expressing in writing, critical thinking, and innovative thinking.

In early stage, EduPCR uses anonymity strategy and all reviewer assignment work is done by computer. Instructors assess and give scores to students based on their performance in coding, reviewing, revising programs, and their abidance to a peer code review process. Thus, collusion is not a problem. In recent years, multi-peer strategy has applied and students' assessment scores for peers' work plays a more important role in determining their final learning performance. The survey data and the interview report indicate that this assessment approach demonstrates high practical values in assessing student learning outcomes in programming languages. Although this approach leads to several interesting research topics for future research in this field [9,15], collusion problem starts to expose.

2) SWoRD: To make peer feedback a viable strategy, SWoRD (Scaffolded Writing and Rewriting in the Discipline) platform is developed, which can effectively and efficiently provide feedback on the aspects of the assignment that are important for instruction [6]. Most saliently, it has algorithms that ensure students take the reviewing task seriously. Instructors can assign many more rich writing-based assignments than they would be able to without being overwhelmed by grading/feedback workload normally [6].

B. Collusion Categories

A categorization of collusion is proposed [10]. It is found that students in online PA systems tend to submit high peer-review scores (sometimes even top scores available) then this kind of behavior is defined as *pervasive collusion*. This behavior makes it harder to identify the top artifacts (an artifact of medium quality may still get as good a score as a top artifact because of grade inflation). Also, it is found that some students gained an unfair advantage by giving higher peer-review scores to students they know. The reason is that students value personal qualities such as friendliness and trust more than mandates on academic conduct. This kind of behavior is defined as *small-circle collusion* [10].

Similarly, we classify collusion into two categories, i.e. *explicit collusion* and *implicit collusion* (see Table I). In the former case, having a prior oral agreement, two or more students give high-score to each other within one task or across more tasks. In the latter case, a tacit agreement, like a latent rule, forms among students on the basis of long-term exchanges. Thus, it can also be named as collective collusion without any prior agreement.

C. Definitions is This Study

For the convenience of understanding, common terms and concepts used in this study are listed in Table I.

TABLE I. DEFINITIONS AND THEIR MEANINGS IN THIS STUDY.

Definition	Meaning
Peer assessment	An arrangement for learners to consider and to specify the level, value, or quality of a product or performance of their peers.
Task	An independent project for students' learning, such as a programming project (code) in EduPCR or a writing composition in SWoRD. In general, there are ten to twelve tasks in one course.
Role	There are three roles in this study: author, reviewer, and instructor. A student plays both the author role and the reviewer role in each task.
Back-evaluation	In SWoRD and EduPCR, after an author receives reviewers' comment or suggestion, back-evaluation to reviewers' review is encouraged (SWoRD) or is required (EduPCR).
High-score threshold	An integer value. A score greater than or equal to it is suspected as a *high-score*. The suggested value is 90.
Explicit collusion	Having a prior agreement, two or more students give *high-score* to each other in one or more tasks.
Implicit collusion	A tacit agreement formed among students on the basis of long-term exchanges among students.
Within-task collusion	It happens within one task, e.g. student A and B give *high-score* to each other within one task. Generally students have explicit agreements and everyone should "pay back" instantly. This kind of collsion is easy to be identified relatively.
Across-task collusion	When some students has a long (explicit or implicit) exchange agreement, they may not "pay back" within a task rather in next tasks. The across-task collusion is relatively hidden so that is not easy to identify.

III. METHODOLOGY

A. Model of Collusion Management in PA

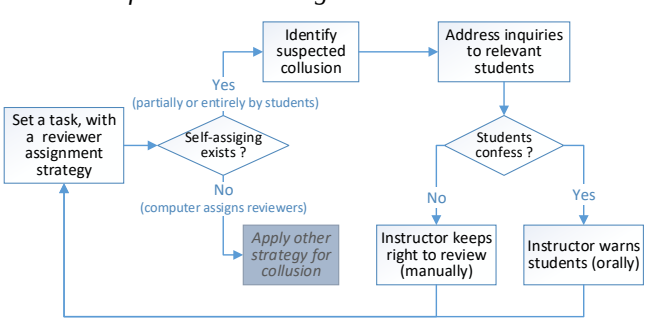

Fig. 1. Model of collusion management in PA

Based on our experience of collusion management in EduPCR, the mutual admiration societies consideration [7, 8], and the collusion categorization [10], we propose a model of collusion management, as shown in Fig. 1. The model is not dedicated to our EduPCR system, it suits to most online PA system in which self-assignment strategy is applied.

In Fig. 1, there are six actions in every programming task, which are explained as follows. At first, an instructor sets a programming project along with a reviewer assignment strategy. If there is no self-assignment case, other collusion management strategy will be applied (not covered in this study). Otherwise, the identification approach will be applied as depicted in the following subsection.

As a result, some students might be suspected for conducting collusion. Then, instructor could address inquiries to them at appointed time. If the students confess the collusion behavior, instructor may warn them. If students do not confess, regardless of students' honesty, instructor keeps the right to review the work (we can name it audit) by relevant students.

Audit can be carried out by comparing the suspected review scores with other peer reviewers' scores, considering with the scoring criteria. If the collusion evidence is strong, a considerable penalty (points deduction) will be given to those student pairs.

However, since penalty is not a preferred choice in education, the actual audit activity by instructor is not strongly suggested unless it is quite necessary. In practice, the inquiry by instructor is sufficient for the majority of students who really conduct collusion. Even so, instructor should tell the students "I keep the right to review your work" before saying goodbye.

B. Collusion Identification Algorithm in the Model

Collusion happens when two or more peer reviewers intentionally give high-scores (e.g., 90% through 100%) to each other in PA. Collusion is common phenomenon in PA process, especially if the reviewer assignment is not completely performed by computer, i.e. authors can select their reviewers. Collusion raises unfairness and decreases the entire learning outcome. PA process should address to this issue. Though automatic random reviewer assignment by computers helps to alleviate the problem, an effective automatic collusion identification mechanism is a much desired feature, especially when the class size is big. In EduPCR, a co-occurrence network approach is developed to identify collusion suspected. The collusion identification algorithm is depicted as follows.

The collusion between two students (no matter who is the reviewer) can happen within a task or across many tasks. The former is named as *within-task collusion*, and the latter is defined as *across-task collusion* (see Table I).

1) Within-task collusion: Suppose N is the total number of students in one class, we define $coMatrix[N][N]$ as the co-occurrence matrix. The element values of the matrix are non-negative integers and are calculated as the following:

- Set a threshold of *high-score*. *High-score* we named is a score equal to or above predefined threshold, e.g. 90%. The value can be configured by course instructor.
- Initialize all matrix elements with value of zero;

- Maintain element value in matrix. Suppose student *i* is one reviewer of student *j*, if *i* gives *j* a *high-score* in review stage and *j* gives *i* a *high-score* in back-evaluation stage, then plus one to the element value of *coMatrix*[*i*][*j*].

2) Enhanced collusion considering across-task cases: Timing is an important factor in collusion identification. We believe that two more adjacent collusion events will have bigger collusion enhancement than two non-adjacent ones. Within-task collusion gets the highest top-ups on enhancement. The formula to compute enhancement of collusion is defined in formula (1), in which c_{ij} is the value of *coMatrix*[*i*][*j*], *W* is a reference constant that denotes the significance of collusion within one task, *k* and *l* are the sequence number of two different tasks. In (1), value of $x^{-abs(k-l)}$ can be considered as a damped exponential [16], in which *x* is an empirical value and *abs*(x) ⩾ 1. The purpose of the damped exponential is that two short-time-separated events will have a bigger collusion enhancement value than two long-time-separated events. In our experiences of this study, a good performance was obtained when *x* is assigned to *e* (2.71828).

$$C_{ij} = C_{ij} + W * x^{-\mathrm{abs(k-l)}} \tag{1}$$

IV. Implementation of the Model

A. Settings in EduPCR

After applying computer-assigning strategy for many academic years, we attempted a strategy including self-assignment cases in our *C Programming* course to explore its motivation effect to students' engagement in PA.

The system was configured as "3 plus 2" mode. That is to say, in each task, computer generated three reviewers for every author randomly which guaranteed fairness. Meanwhile, each author was also allowed to choose two additional reviewers after he/she submitted source code. It was based on the characteristic that the collusion phenomenon will be more obvious to some extend when authors were allowed to choose their favorite reviewers. The choice of "3" ensured the randomness of reviewer selection, and the choice of "2" gave students push of finishing their work as soon as possible, however, it might leave space for students to conduct collusion.

B. Identification Results of Suspected Collusion

At the beginning of this course, students were told that collusion was unacceptable as a misconduct and student with a confirmed collusion would get a heavy penalty (score deduction).

Based on the identification algorithm, EduPCR ran the collusion identification for the class after every task. The collusion results were shown as co-occurrence diagrams depicted in Fig. 2 and Fig. 3. For protecting students' privacy, all nodes in these two figures were labeled with generated IDs within EduPCR system.

In Fig. 2, only within-task collusion was considered and *W* was set to zero. The pairs of students, such as (6374, 6371) and (6383, 6381) showed up in Fig. 2 as suspects of collusion but their enhancement of collusion did not increase in Fig. 3. The evidence of collusion was very weak for these students.

In Fig. 3, the value of *W* in (1) was set to *5* and cross-task collusion was incorporated in the enhancement calculation. The visualization effect of collusion was enhanced after computing cross-task collusion.

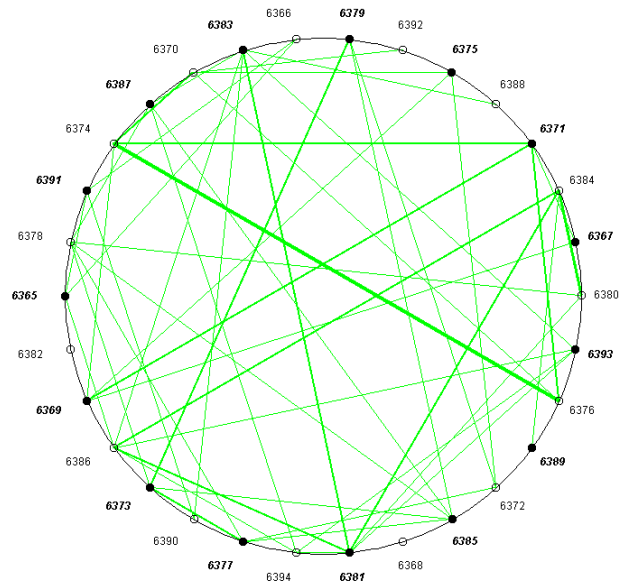

Fig. 2. Co-occurrence diagrams of suspected collusion without enhancement

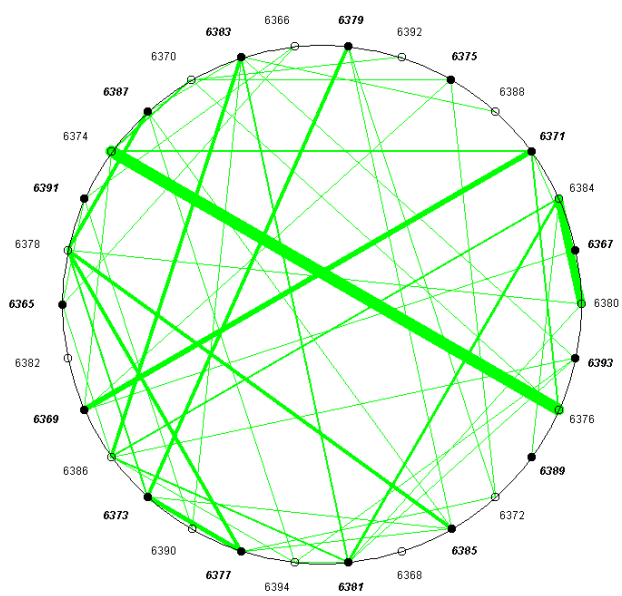

Fig. 3. Co-occurrence diagrams of suspected collusion after enhancement

The enhancement of collusion became stronger after considering cross-task collusion, as shown in Fig. 3. In Fig. 3, student pairs of (6374, 6376), (6384, 6380), and (6369, 6371) had top three highest collusion possibility. As proposed in the management model depicted in Fig. 1, these students would be inquired and possibly warned. Furthermore, they were informed that instructor keeps the right to review their work and to punish them.

V. CONCLUSION AND FUTURE WORK

An empirical study conducted in a university-level *C Programming* course shows that the proposed model of management is reasonable and the implementation description helps related scholars to understand how to apply the identification algorithm in diverse PA contexts.

Even though we have designed some stages such as inquiries to collusion suspects and warnings to confessed students, it is not suggested to utilize them often. In our understanding, education is not only for the dissemination of knowledge and skills, but also for the cultivation of ideology and morality. Instructor should tolerate students' mistake such as collusion. The importance should be given to helping them correct it and building up their healthy personality.

In near future, some valuable work could be concerned by relevant scholars in PA.

1) Experiments in large scale are expected: Larger scale experiments are suggested to explore the essence of collusion and to verify the feasibility of identification algorithm. Actually, the implementation in this study could be regarded as a testing of protocol. If the implementation can be conducted in large-size classes, the finding might be more accurate and other valuable knowledge might be acquired.

2) The identification algorithm could be refined: For example, we considered the frequency of giving *high-score* in this study, but the degree of "high" was not measured. Although 90 and 100 are both in the set of *high-score*, they have different weight when we judge how much likely a rating action is to be regarded as a suspected collusion behavior.

REFERENCES

[1] J. S. Kane and E. E. Lawler, "Methods of peer assessment," Psychological Bulletin, vol. 85, no. 3, pp. 555–586, 1978.

[2] K. J. Topping, "Peer assessment," Theory into Practice, vol. 48, no. 1, pp. 20–27, 2009.

[3] N. Falchikov, "Peer feedback marking: Developing peer assessment," Innovations in Education and Training International, vol. 32, no. 2, pp. 175–187, 1995.

[4] M. Freeman, "Peer assessment by groups of group work," Assessment and Evaluation in Higher Education, vol. 20, no. 3, pp. 289–300, 1995.

[5] P. Vickerman, "Student perspectives on formative peer assessment: An attempt to deepen learning?" Assessment & Evaluation in Higher Education, vol. 34, no. 2, pp. 221–230, 2009.

[6] C. D. Schunn, "Writing to learn and learning to write through SWoRD," in Adaptive Educational Technologies for Literacy Instruction, S.A. Crossley and D. S. McNamara, Eds. New York: Taylor & Francis, Routledge, 2016, pp. 243-259.

[7] P. A. Carlson, F. C. Berry, and D. Voltmer, "Incorporating student peer-review into an introduction to engineering design course," In Proceedings of 35th Annual Conference of Frontiers in Education (pp. F2C20-F2C25). IEEE, 2005.

[8] X. Li , "Incorporating a code review process into the assessment," In the 20th Annual Conference of the National Advisory Committee on Computing Qualifications, Nelson, New Zealand, pp. 125–131, 2007.

[9] Y. Wang, W. Ai, Y. Liang, and Y. Liu, "Towards motivating participants to assess peers' work more fairly: Taking programming language learning as an example," Journal of Educational Computing Research, vol. 52, no. 2, pp. 180−198, 2015.

[10] Y. Song, Z. Hu, and E. F. Gehringer, "Collusion in educational peer assessment: How much do we need to worry about it?" Frontiers in Education Conference (pp. 1-8), Indianapolis, IN, USA, IEEE, 2017.

[11] R. M. Blank, "The effects of double-blind versus single-blind reviewing: Experimental evidence from the American Economic Review," The American Economic Review, vol. 81, no. 5, pp. 1041−1067, December, 1991.

[12] R. Ballantyne, K. Hughes, and A. Mylonas, "Developing procedures for implementing peer assessment in large classes using an action research process," Assessment & Evaluation in Higher Education, vol. 27, no. 5, pp. 427−441, 2002.

[13] Y. Wang and F. Sun, "How to choose an appropriate reviewer assignment strategy in peer assessment system? Considering fairness and incentive," In proceeding of 2018 Annual International Conference on Education and Social Development (ESD'2018), May 18th-20th, 2018, Xi'an, China. (in press).

[14] P. M. Papadopoulos, T. D. Lagkas, and S. N. Demetriadis, "How to improve the peer review method: Free-selection vs assigned-pair protocol evaluated in a computer networking course," Computers & Education, vol. 59, no. 2, pp. 182-195, 2012.

[15] Y. Wang, H. Li, Y. Feng, Y. Jiang, and Y. Liu, "Assessment of programming language learning based on peer code review model: Implementation and experience report," Computers & Education, vol. 59, no. 2, pp. 412−422, 2012.

[16] N. Kannan and D. Kundu, "Estimating parameters in the damped exponential model," Signal Processing, vol. 81, no. 11, pp. 2343-2351, 2001.